Então, vamos colocar em prática os conceitos de condições lógicas que estudamos previamente, usando AND, OR, NOT e assim por diante. Começaremos criando um novo script SQL e nos certificando de que estamos com a base de dados "sucos\_vendas" selecionada.

Nossa primeira consulta será simples, com uma condição que já aprendemos como trabalhar:

Texto

O conteúdo gerado por IA pode estar incorreto.

Ao executar o script, você pode verificar na coluna "SABOR" que todos os registros encontrados têm valor "manga". Então, vamos incrementar nossa consulta, adicionando mais uma expressão à condição:



Com essa seleção, vamos filtrar apenas os registros que tenham sabor manga **ou** tamanho de 470 ml (**ou** até mesmo os dois). Executando o script, podemos reparar que o primeiro item do resultado atende às duas expressões; os quatro produtos seguintes têm sabor manga, porém tamanhos diferentes de 470 ml; e o último registro tem 470 ml, no entanto, é sabor laranja.

A seguir, criaremos uma consulta com AND:



Nesse caso, como usamos o AND, o retorno terá apenas um registro - nessa tabela, é o único produto cujo sabor é manga **E\*\* o tamanho é igual a 470 ml \*\*ao mesmo tempo**.

Agora, vamos inserir o NOT, para fazer a seleção inversa: uma consulta de todos os registros **exceto** os que têm sabor manga **e\*\* tamanho 470 ml \*\*ao mesmo tempo**:

**SELECT** \* **FROM** tabela\_de\_produtos **WHERE** **NOT** (SABOR = 'Manga' **AND** TAMANHO = '470 ml');

Dessa consulta obteremos um resultado bem extenso. Já se colocarmos o NOT em combinação com OR, o retorno não será tão abrangente:

**SELECT** \* **FROM** tabela\_de\_produtos **WHERE** **NOT** (SABOR = 'Manga' **OR** TAMANHO = '470 ml');

Desse script resulta uma seleção em que não veremos nenhum registro com sabor manga e nenhum registro com tamanho 470 ml.

Há ainda outras formas de usar o NOT. Podemos, por exemplo, inseri-lo na frente de apenas uma das expressões:

**SELECT** \* **FROM** tabela\_de\_produtos **WHERE** SABOR = 'Manga' **AND** **NOT** (TAMANHO = '470 ml');

Nesse caso, procuraremos itens que tenham sabor manga **E\*\* que o tamanho \*\*não** seja 470 ml. Isso é, ao rodar o script, na tabela resultante você encontrará todos os produtos com sabor manga **exceto** os que tem tamanho 470 ml - qualquer outro tamanho será apresentado.

Existem mais condições lógicas além das que expliquei no vídeo anterior. Uma delas é o IN, que pode ser interpretado como "contido":



Com esse script, estamos selecionando todos os produtos da "tabela\_de\_produtos" cujo sabor está **contido** na lista ('Laranja', 'Manga'). Essa consulta é equivalente à seguinte:



Ou seja, também procuramos registros cujo sabor seja laranja ou manga.

Além disso, temos a opção de criar condições que mesclam diferentes colunas. Para exemplificar, vamos fazer outra consulta com o operador IN, desta vez com a tabela de clientes:

**SELECT** \* **FROM** tabela\_de\_clientes **WHERE** CIDADE **IN** ('Rio de Janeiro', 'São Paulo') **AND** IDADE >= 20;

Rodando esse script, teremos como resultado uma tabela com todos os clientes cuja cidade consta como Rio de Janeiro **ou** como São Paulo, **e** cuja idade seja maior ou igual a 20.

Ademais, há sempre a possibilidade de criar consultas mais complexas:

**SELECT** \* **FROM** tabela\_de\_clientes **WHERE** CIDADE **IN** ('Rio de Janeiro', 'São Paulo') **AND** (IDADE >= 20 **AND** IDADE <= 22);

É interessante fazer uso dos parênteses, como em (IDADE >= 20 AND IDADE <=22), para manter a ordem em seu script. Mesmo que não acarrete diferenças na execução da consulta, trata-se de uma boa prática de organização. Nesse caso, estamos unindo condições que dizem respeito ao mesmo campo. Um código bem indentado e com uma boa estrutura facilita bastante na hora de ler, interpretar e corrigir problemas.

Assim, nesse vídeo buscamos colocar em prática toda a teoria que aprendemos anteriormente, aplicando os operadores e sinais lógicos em uma variedade de consultas.